5. DETAILED REQUIREMENTS

TACO2 comprises the following protocol layers: the TACO2 NRTS, NETBLT, IP, Header Abbreviation, FEC, and either HDLC or SLIP. Detailed requirements for those protocol layers are provided in this section.

5.1 NITFS reliable transfer server for TACO2 (TACO2 NRTS). The TACO2 NRTS shall provide a (locally-defined) sending and receiving interface to the NITFS user process. It shall transfer (and if necessary act upon) the message and the metamessage to its peer TACO2 NRTS via NETBLT, and it may pass a destination IP address (at the message source) and exchange a set of parameters with NETBLT. The TACO2 NRTS described here assumes an active sender and a passive receiver ("push" operation); as of the effectivity date (Effectivity 4) the TACO2 NRTS also shall support an active receiver and passive sender ("pull" operation).

5.1.1 Metamessage definition. The metamessage shall be used to communicate information about the NITFS message between the source and the destination system (and intermediate systems, if any). It shall be transferred along with the NITFS message itself by TACO2, in a manner that allows the receiver to make an unambiguous connection between message and metamessage. In duplex communications, a return metamessage shall be transferred from receiver to sender, to allow possible negotiation or confirmation of some component values as specified in 5.1.1.2. The notation for the information about the message is designed to allow for additional attributes as the need arises, but with a minimum of additional complexity.

5.1.1.1 Description. The metamessage shall be a null-terminated string of 8-bit American Standard Code for Information Interchange (ASCII) letters, numbers, and some special characters (MIL-STD-2500 for the definition of the acceptable set of ASCII). The maximum length of the metamessage is unbounded (although implementations may have a limit on the length they can accept; this limit shall be at least 255 characters, and characters beyond the limit may be discarded), and the end of the metamessage shall be denoted by the occurrence of a null byte (hexadecimal 0x00). The individual components of the metamessage shall have the form:

<component-name>=<component-value>

where:

a. <component-name> shall be chosen from the list of components (see 5.1.1.2)

b. "=" is the literal ASCII character

c. <component-value> is a sequence of ASCII characters, not including the space or comma characters, providing the value for that component (numbers shall be given in ASCII decimal representation).

Components shall be separated by spaces or commas. The maximum length of the name components (message name, file name, sender name, and recipient name) is unbounded, although implementations may have a limit on the length they can accept; this limit shall be at least 32 characters for each such component. The only exception to the component form given above is the mandatory first component, Version, which shall be three bytes with hexadecimal value 0x5E0101. The number of occurrences of components that may occur multiple times is unbounded, but may be limited by the acceptable length of the metamessage. Components not specifically permitted to occur multiple times in a single metamessage shall occur no more than once in a metamessage.

5.1.1.2 Components. The following is a list of the metamessage components that may be supplied, and shall be recognized, by TACO2. Each component description includes an indication of whether it is optional or mandatory, its default value if it is absent, its form, and its meaning. Except for the mandatory initial component Version, the sequence of components in the metamessage is immaterial. Table I summarizes the metamessage components.

TABLE I. Metamessage components.

NAME	FORMAT	PRESENCE	DEFAULT
Version	0x5E0101	Mandatory	
Message Name	MNAME=<message-name>	Mandatory	None
File Name	FNAME=<file-name>	Optional	None
Message Type	TYP=<message-type>	Optional	NTF
Message Length	LEN=<message-length>	Optional	None
Start Point	STRT=<start-point>	Optional	0
Sender Name	FROM=<name>	Optional	None
Recipient Name	TO=<name>	Optional	None
Validity	NVAL=<first-octet>:<last-octet>	Optional	All valid
Criticality	CRIT=<first-octet>:<last-octet>	Optional	All equal

Geolocation	GEOG=<ddmmss><N|S><dddmmss><E|W>	Optional	None

5.1.1.2.1 Version. (Mandatory). This shall be the first three bytes of the metamessage, and shall be hexadecimal 0x5E0101. The particular representation was chosen for compatibility with possible future alternative Abstract Syntax Notation One (ASN.1) representations of the metamessage.

5.1.1.2.2 Message name. (Mandatory, must occur within the first 255 characters of the metamessage). No default. MNAME=<message-name>, where <message-name> shall be a unique identifier for the NITFS message. (May be used for accountability tracking in some systems.) The <message-name> shall be unique for communication between two systems, as it will be used to identify (and possibly combine) multiple copies of the same message and to resume interrupted transfers (see 5.1.1.2.6). The originator may generate <message-name> with any method that meets the uniqueness requirement; file modification date-time may be included to help ensure uniqueness.

5.1.1.2.3 File name. (Optional). No default. FNAME=<file-name>, where <file-name> shall be a name under which the message may be stored in the destination system (it might be the name under which the originator stored it). The destination is under no obligation to use this name. If the receiver includes this component in a metamessage returned to the sender, <file-name> shall be the name under which the message will be stored in the receiving system.

5.1.1.2.4 Message type. (Optional). Default is NTF, signifying a NITF message. TYP=<message-type>, where <message-type> shall be the kind of message being transferred. NOTE: Other message types, such as text and spreadsheets, are To Be Resolved (TBR).

5.1.1.2.5 Message length. (Optional). No default. LEN=<message-length>, where <message-length> shall be a decimal number, represented in ASCII, describing the number of bytes in the NITFS message.

5.1.1.2.6 Start point. (Optional). Default = 0. STRT=<start-point>, where <start-point> shall be the decimal offset, represented in ASCII, of the first octet proposed to be transferred. (The octets in a file are numbered starting with 0). This may be used to resume interrupted transfers. If the receiver includes this component in a metamessage returned to the sender, <start-point> shall be the offset from the first octet in the file to be transferred, which shall be less than or equal to that proposed by the sender. In this case, the sender shall use the returned value as the actual start point of the transfer.

5.1.1.2.7 Sender name. (Optional). No default. FROM=<name>, where <name> shall be a meaningful ASCII name of the sender.

5.1.1.2.8 Recipient name. (Optional). No default. TO=<name>, where <name> shall be a meaningful ASCII name of a recipient. This component may occur multiple times in a single metamessage.

5.1.1.2.9 Validity. (Optional). Default is to assume that the entire message is valid. NVAL=<first-octet>:<last-octet>, where <first-octet> and <last-octet> shall be the inclusive boundaries of a possibly invalid portion of the message. The first octet of a message is numbered 0. This component may occur multiple times in a single metamessage. The receiver may use this information to assist in message interpretation.

5.1.1.2.10 Criticality. (Optional). Default is to assume that the entire message is equally critical. CRIT=<first-octet>:<last-octet>, where <first-octet> and <last-octet> shall be the inclusive boundaries of a portion of the message which is critical to the interpretation of the overall message. The first octet of a message is numbered 0. This component may occur multiple times in a single metamessage. Special actions may be taken to increase the probability of correct reception of this portion of the message (see 5.2.8.1.1).

5.1.1.2.11 Geolocation. (Optional). No default. GEOG=<ddmmss><N|S><dddmmss><E|W>, with coordinates in the Geographic Coordinate System. The coordinate shall refer to the center of the image in the message. The first element represents degrees, minutes, and seconds of latitude; the second is N (north) or S (south); the third is degrees, minutes, and seconds of longitude; and the last is E (east) or W (west).

5.1.1.2.12 Additional components. Other components, with the same form, are reserved for future versions of TACO2.

5.1.1.2.13 Unrecognized components. An unrecognized component (such as, one that does not appear on this list) may be discarded by the recipient, and shall not disrupt the protocol exchange.

5.1.2 TACO2 NRTS operation. This section describes the interaction between the TACO2 NRTS and NETBLT. The details of the interaction and interfaces between protocol layers are not specified herein; however, the complete TACO2 protocol stack shall interact with other TACO2 protocol stack implementations in a manner consistent with this description. The descriptions in this section are supplemented with time sequence diagrams, which illustrate the relationship between operations across the TACO2 NRTS to NETBLT interface. The vertical lines represent the interfaces, and time increases down the diagram.

5.1.2.1 Summary of operation. When a message is to be transferred, the TACO2 NRTS requests that NETBLT open a connection to the intended receiver. Once the connection is opened successfully, the message is transferred between the TACO2 NRTS and NETBLT as a series of buffers at both the sending and receiving end. All buffers except the last one shall be the same size. When the transfer is complete, the NETBLT connection is closed. If the entire message is not transferred successfully, the sending TACO2 NRTS may attempt to open a new connection and restart the transfer (see 5.1.1.2.6). Status information not described here may be provided by NETBLT to the TACO2 NRTS, and by the TACO2 NRTS to the user process.

5.1.2.2 Connection establishment. The sequence of events by which the TACO2 NRTS establishes a connection via NETBLT is shown on figure 5.

FIGURE 5. TACO2 NRTS connection establishment.

5.1.2.2.1 Sending end system.

5.1.2.2.1.1 Connection-Request. The TACO2 NRTS shall request that NETBLT open a connection to the intended receiver with a Connection_Request, which has the following parameters:

a. Foreign host. A 32-bit IP address for the destination of the NITFS message.

b. Foreign port. Shall be 1.

c. Checksum flag. Shall require data checksumming.

d. Send/Receive flag. Shall indicate send.

e. Metamessage. Shall be as specified in 5.1.1

f. Proposed maximum number of outstanding buffers.

g. Proposed buffer size.

5.1.2.2.1.2 Connection-Response. NETBLT shall respond to the Connection_Request from the TACO2 NRTS with Connection_Response, which has two parameters:

a. Success/failure flag.

b. Return metamessage, if successful; reason for failure, if unsuccessful.

5.1.2.2.2 Receiving end system.

5.1.2.2.2.1 Listen-Request. When the TACO2 NRTS is prepared to receive a message, it shall invoke NETBLT with a Listen_Request, which has the following parameters:

a. Local port. Shall be 1.

b. Maximum acceptable number of outstanding buffers.

c. Maximum acceptable buffer size.

5.1.2.2.2.2 Connection-Request-Received. When a connection is opened by a foreign host, NETBLT shall respond to the Listen_Request of the receiving-side TACO2 NRTS with Connection_Request_Received, which has the following parameters:

a. Foreign host. The 32-bit IP address of the source of the NITFS message.

b. Send/receive flag. Should indicate send (by the other end).

c. Metamessage. Shall be the metamessage provided by the sending TACO2 NRTS.

5.1.2.2.2.3 Connection-Request-Response. The TACO2 NRTS shall respond to NETBLT with Connection_Request_Response, which has two parameters:

a. Accept/reject flag.

b. Return metamessage, as specified in 5.1.1, if successful; reason for rejection, if unsuccessful.

5.1.2.3 Data transfer. The sequence of actions by which the TACO2 NRTS transfers a message via NETBLT is shown on figure 6. This sequence shall be repeated until all data in the message have been transferred, unless the transfer is terminated prematurely.

FIGURE 6. TACO2 NRTS data transfer.

5.1.2.3.1 Sending end system.

5.1.2.3.1.1 Send-Buffer-Setup. The sending NETBLT shall request the next buffer full of data from the TACO2 NRTS with Send_Buffer_Setup, which has two parameters:

a. Buffer number, where buffers are numbered consecutively starting with 0.

b. Requested size of the buffer.

5.1.2.3.1.2 Send-Buffer-Response. The TACO2 NRTS shall respond to NETBLT with Send_Buffer_Response, which has the following parameters:

a. Identification of the buffer with data to be sent. The data in that buffer shall be consecutive bytes of the message to be transferred, starting with byte number ((buffer number-1)*(requested size of the buffer)), where bytes are numbered starting with 0.

b. Actual size of the buffer.

c. Last buffer flag. Set if this is the last buffer of the message.

5.1.2.3.1.3 Send-Flush-Buffer. When NETBLT has finished with the buffer, it shall invoke Send_Flush_Buffer, which has one parameter: identification of the buffer.

5.1.2.3.2 Receiving end system.

5.1.2.3.2.1 Receive-Buffer-Setup. The receiving NETBLT shall request an empty buffer from the TACO2 NRTS with Receive_Buffer_Setup, which has two parameters:

a. Buffer number.

b. Requested size of the buffer.

5.1.2.3.2.2 Receive-Buffer-Response. The TACO2 NRTS shall respond to NETBLT with Receive_Buffer_Response, which has one parameter: identification of the buffer.

5.1.2.3.2.3 Receive-Flush-Buffer. When NETBLT has filled the buffer, it shall invoke the TACO2 NRTS with Receive_Flush_Buffer, which has the following parameters:

a. Identification of the buffer. The data in that buffer shall be consecutive bytes of the message to be transferred, starting with byte number ((buffer number-1)*(requested size of the buffer)), where bytes are numbered starting with 0. Received buffers may be returned out of order by NETBLT; the correct placement of data in the received message shall be established by the TACO2 NRTS, by correctly associating the buffer identification and the buffer number.

b. Actual size of the buffer.

c. Last buffer flag. Set if this is the last buffer of the message.

5.1.2.4 Connection termination. A TACO2 message transfer may be terminated by NETBLT or by the TACO2 NRTS. The possible terminations are shown on figure 7.

FIGURE 7. TACO2 NRTS connection termination.

5.1.2.4.1 NETBLT-invoked termination.

5.1.2.4.1.1 Close. After all buffers comprising the message have been transferred, NETBLT shall notify both the sending and receiving NRTS with a Close, which has one parameter: List of possibly invalid bytes (this parameter is only meaningful in receive simplex operation; in any other operation, it shall be empty).

5.1.2.4.1.2 Exception-Report. If NETBLT is unable to complete the transfer, it shall send an Exception_Report to the TACO2 NRTS. Exception_Report has one parameter: Reason for the report.

5.1.2.4.2 TACO2 NRTS-invoked termination.

5.1.2.4.2.1 Quit-request. The TACO2 NRTS shall invoke Quit_Request to cause an orderly but premature termination of a TACO2 message transfer. Quit_Request has one parameter: Reason for request.

5.1.2.4.2.2 Abort-request. The TACO2 NRTS shall invoke Abort-Request to cause an immediate termination of a TACO2 message transfer. Abort_Request has one parameter: Reason for request.